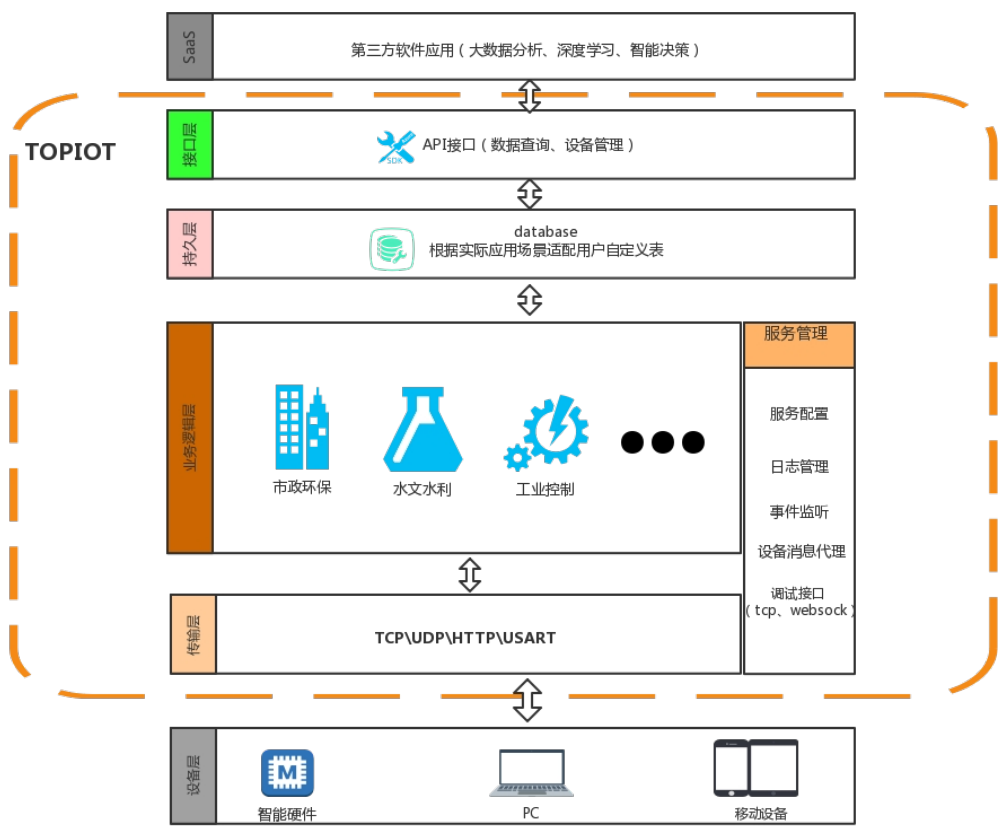


一、TOPIOT 平台概述

spider (qq:799509033, email:xizhonghuai@foxmail.com)

TOPIOT 物联网平台是面向个人、企业开发者提供智能硬件互联的一站式云服务平台。平台提供了从设备接入（多协议适配）、业务逻辑处理、数据持久化、数据推送、到在线调试等全生命周期服务的能力。

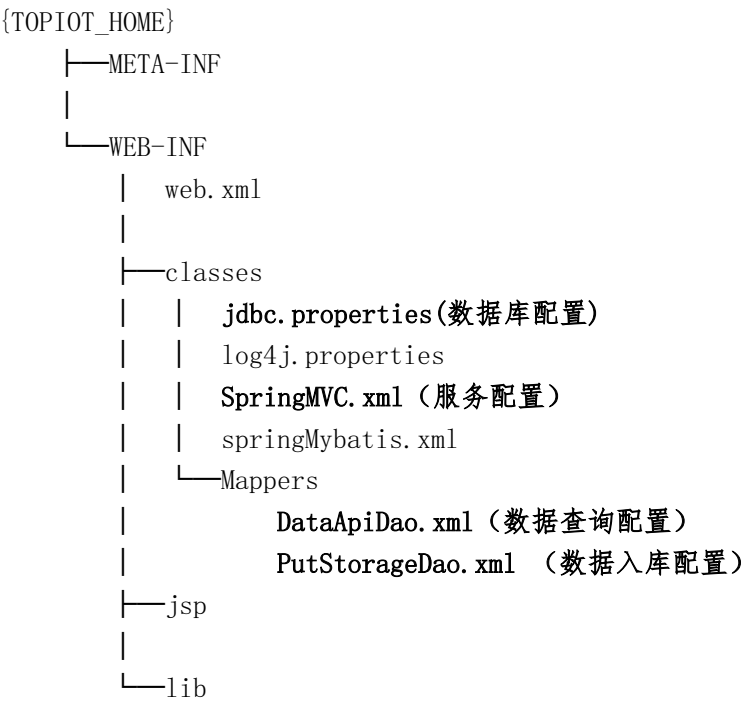
开发者通过该平台可轻松实现设备互连，开发者只需关注其业务逻辑，可最大限度缩短物联网项目研发周期。



图表 1 平台架构

二、服务配置说明

1、TOPIOT 配置文件目录结构



2、服务配置

场景：新增测试服务，tcp 协议，端口：1234，实时表 datatemp 结构

如下：

字段	数据类型	说明
Id	Integer	序号，自动递增
deviceID	String	设备唯一标识
value	float	采集数据
dt	dateTime	采集时间

step1：服务配置

打开 “SpringMVC.xml” 文件，复制以下代码，粘贴至服务配置区(xml 注释)，

```
<bean id="test" class="com.transportlayer.Tcp">
    <property name="serviceId" value="test"></property>
    <property name="Handler" value="com.protocol.test.TestHandler"></property>
```

```

        <property name="port">
            <list>
                <value>1234</value>
            </list>
        </property>
        <property name="idle" value="600"></property>
        <property name="decodecharset" value="ISO-8859-1"></property>
        <property name="encodecharset" value="ISO-8859-1"></property>
        <property name="pushUrl" value="http://127.0.0.1/push"></property>
        <property name="pushflag">
            <value>true</value>
        </property>
        <property name="debugflag">
            <value>true</value>
        </property>
        <property name="toServiceId" value="test"></property>
    </bean>

```

代码说明：

- 1)、id: 服务唯一标识，由字母或数字组成。
- 2)、class: 传输层协议，取值如下：
 - tcp: com.transportlayer.Tcp
 - udp: com.transportlayer.Udp
 - usart: com.transportlayer.Usart
- 3)、serviceId: 服务唯一标志，同id。
- 4)、Handler: 应用层协议(业务逻辑部分，取值为包+类名):
 - 例: com.protocol.test.TestHandler (TestHandler类为业务逻辑部分)
- 5)、port: tcp、udp端口，支持多个配置。
- 6)、comName: 串口端口号。
- 7)、baud: 串口波特率。
- 8)、decodecharset、encodecharset: 接收、发送数据编码方式。
- 9)、idle: 数据读写通道空闲时间，单位秒。仅对tcp有效。
- 10)、pushUrl: 数据推送地址。
- 11)、pushflag: http推送开关，开启后，设备端数据将转发到pushUrl上。

推送协议: 服务唯一标识ID(固定长度16个字节，不足空格补)+设备id(固定长度16个字节，不足空格补)+数据

12)、debugflag: 调试开关，设置为true可通过以下协议与设备进行数据交互，支持websocket、tcp。

websocket地址: ws://127.0.0.1:8081/topiot/ws

tcp地址: 127.0.0.1:10401

客户端→设备: 服务唯一标识ID(固定长度16个字节，不足空格补)+设备id(固定长度16个字节，不足空格补)+数据

设备→客户端: 服务唯一标识ID(固定长度16个字节，不足空格补)+设备

id(固定长度16个字节，不足空格补)+数据

13)、toServiceId: 默认值为自身服务id，表示该服务下设备与设备之间可实现单工通信。若配置为其他服务id，则能实现不同服务下的设备与设备之间通信。

通信协议: M2M+(13 个空格)+设备 id(固定长度 16 个字节，不足空格补)+
数据

step2: 入库配置

打开“PutStorageDao.xml”文件，复制以下代码粘贴至 sql 配置区（xml 注释）

```
<insert id=" insertTestRealtimeData " parameterType="com.protocol.test.TestDataModel">
    insert into datatemp (deviceID, value, dt) values
    ({regId},{value},{tm})
</insert>
```

代码说明:

1)、id、parameterType的值通过api接口获取，相关api接口见第五节获取应用层协议信息。

step3: 重启服务

3、通过后台管理系统配置服务

The screenshot shows a web interface for configuring services. On the left is a sidebar with a menu containing 'Service List' (服务列表), 'New Service' (新建服务), and 'Service Test' (服务测试). The main area is titled 'Add Service' (添加服务) and contains a form with the following fields: 'Service ID' (服务id) with a placeholder 'Service unique identifier, composed of letters and numbers'; 'Application Protocol' (应用层协议) set to 'Test' (测试); 'Transport Protocol' (传输层协议) set to 'TCP'; 'Listen Port' (监听端口) with a placeholder 'Multiple ports separated by commas'; 'Idle Timeout' (空闲超时时间) set to '600'; 'Character Encoding' (字符编码方式) and 'Character Decoding' (字符解码方式) both set to 'ISO-8859-1'; 'Data Forwarding' (数据转发) and 'Test Service' (测试服务) as toggle switches. A 'Submit' (提交) button is at the bottom of the form.

图表 2 服务配置

服务列表								
详细信息								
服务id	状态	传输层协议	端口	调试端口	应用层协议	描述	版本	操作
wxy	运行	TCP	[5011]	true	wxy.WxyHandler	危险源监控预警系统	V1.0	停止
wxydebug	运行	TCP	[1105]	false	wxy.WxyDebugHandler	危险源监控预警系统调试服务	V1.0	停止
json	运行	TCP	[4010]	true	jsonrtu.JsonRtuHandler	简易RTUjson数据传输	V1.0	停止

图表 3 服务运行情况

二、在线调试接口

平台提供 tcp、ws 接口实现与设备远程数据交互，在服务配置 xml 代码中，需将“debugflag”属性设置为 true。服务会自动转发设备数据至 tcp、ws 客户端。同时客户端也可发送数据至指定设备。

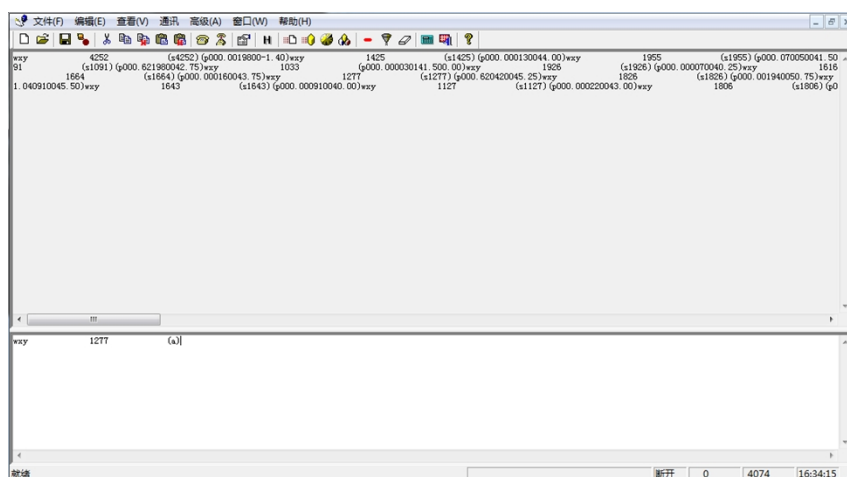
tcp 服务地址：127.0.0.1:10401

ws 服务地址：ws://127.0.0.1:8081/topiot/ws

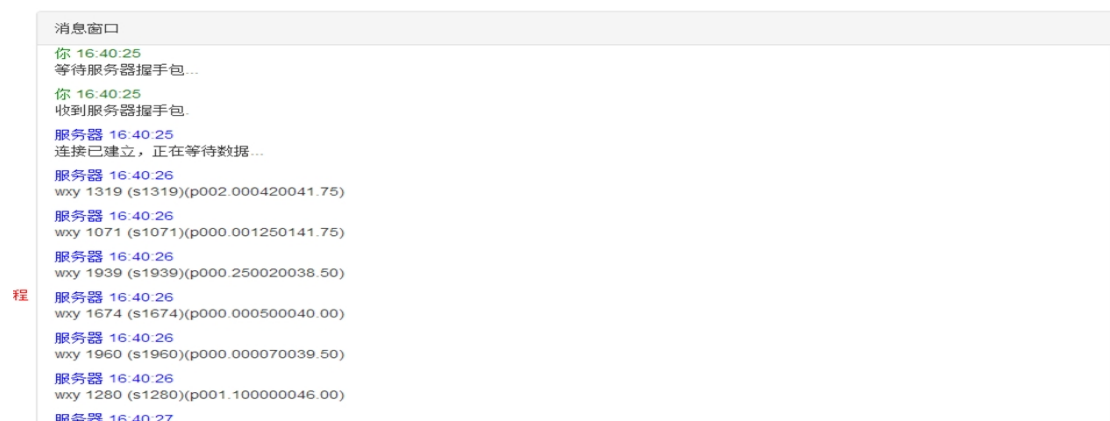
转发协议：

客户端→设备：服务唯一标识ID(固定长度16个字节，不足空格补)+设备id(固定长度16个字节，不足空格补)+数据

设备→客户端：服务唯一标识ID(固定长度16个字节，不足空格补)+设备id(固定长度16个字节，不足空格补)+数据



图表 4 tcp 连接



图表 5 ws 连接

三、数据推送

在服务配置 xml 代码中，配置“pushUrl”的值为一个能正常访问的 url，支持任意端口，同时将“pushflag”设置为 true，服务会自动转发设备数据至 url 上。该功能需要有自己的服务器资源。

推送协议：

服务唯一标识ID(固定长度16个字节，不足空格补)+设备id(固定长度16个字节，不足空格补)+数据

四、设备消息代理

在服务配置 xml 代码中，配置“toServiceId”的值为一个有效的 serviceId，终端设备 A 可通过平台向服务 id 为 serviceId 下的任意设备实现数据通信。需特别说明终端设备 A 只能发送数据给对方。

代理协议：

M2M+(13 个空格)+设备 id(固定长度 16 个字节，不足空格补)+数据

五、API 接口说明

1、获取指定服务信息

简要描述：

获取指定服务信息

请求 URL:

http://127.0.0.1:8081/topiot/service.api/getservice

请求方式:

post/get

参数:

参数名	必选	类型	说明
serviceId	否	string	要获取服务的 id, 若不填则获取全部

返回示例

```
{
  "reason" : "success",
  "result" : {
    "0" : {
      "debugflag" : "true",
      "handler" : "com.protocol.wxy.WxyHandler",
      "handlerMark" : "危险源监控预警系统",
      "handlerVversion" : "V1.0",
      "port" : "[5011]",
      "pushUrl" : "http://127.0.0.1/push/",
      "pushflag" : "false",
      "serviceId" : "wxy",
      "status" : "true",
      "toServiceId" : "wxy",
      "transport" : "TCP"
    },
  }
}
```

返回参数说明

参数名	说明
-----	----

参数名	说明
debugflag	调试端口是否开启
handler	应用层协议
handlerMark	应用层协议描述
handlerVversion	应用层协议版本号
port	监听端口
pushUrl	数据推送地址
pushflag	是否开启推送
serviceId	服务 id
status	服务状态，true 为启动
toServiceId	M2M 消息代理服务 id
transport	传输层协议

2、启动、停止指定服务

简要描述：

启动、停止指定服务

请求 URL：

http://127.0.0.1:8081/topiot/service.api/controlservice

请求方式：

post/get

参数：

参数名	必选	类型	说明
serviceId	是	string	服务 id
cmd	是	string	stop/start

返回示例

```
{
  "reason" : "success",
  "result" : "启动成功"
}
```


3、获取指定服务终端信息

简要描述：

获取指定服务终端信息

请求 URL：

http://127.0.0.1:8081/topiot/service.api/getserviceconn

请求方式：

post/get

参数：

参数名	必选	类型	说明
serviceId	是	string	服务 id

返回示例

```
{
  "reason" : "success",
  "result" : {
    "0" : {
      "activityTime" : "1532702719399",
      "address" : "/117.132.196.18:57161",
      "regId" : "4065",
      "registerTime" : "1532702658119"
    },
    ...
    "456": {
      "activityTime" : "1532702969459",
      "address" : "/223.104.254.54:64403",
      "regId" : "1030",
      "registerTime" : "1532702658757"
    }
  }
}
```

返回参数说明

参数名	说明
registerTime	设备首次连接服务器时间戳
activityTime	设备最近活动时间戳
regId	设备唯一标识
address	设备地址

4、获取服务连接数

简要描述：

获取服务连接数

请求 URL：

http://127.0.0.1:8081/topiot/service.api/getservicecount

请求方式：

post/get

参数：

参数名	必选	类型	说明
serviceId	是	string	服务 id

返回示例

```
{
  "reason" : "success",
  "result" : "474"
}
```

5、获取应用层协议信息

简要描述：

获取应用层协议信息

请求 URL：

http://127.0.0.1:8081/topiot/service.api/getprotocol

请求方式：

post/get

参数：

参数名	必选	类型	说明
handler	是	string	协议类名称

返回示例

```
{
  "reason" : "success",
  "result" : {
    "dataModelClassName" : "com.protocol.test.TestDataModel",
    "dataModelProperty" : {
      "0" : "id_序号_String",
      "1" : "regId_设备唯一标识_String",
      "2" : "value_实时数据_Float",
      "3" : "tm_数据上报日期_Date",
    },
    "mark" : "测试服务",
    "putStorageMethod" : {
      "0" : "insertTestRealtimeData",
      "1" : "updateTestRealtimeData",
    },
    "version" : "V1.0"
  }
}
```

返回参数说明

参数名	说明
dataModelClassName	请求协议若存在数据入库，则该参数值表示需入库数据类名称
dataModelProperty	入库数据类属性：参数名_参数描述_参数类型
mark	协议描述
putStorageMethod	入库方法
version	协议版本

备注 数据入库配置中需用到该接口返回参数，相关说明如下：

若某服务应用层协议类名为该接口请求参数值，则入库配置代码为：

```
<insert id=" insertTestRealtimeData " parameterType="com.protocol.test.TestDataModel">
    insert into datatemp (deviceID, value, dt) values ({regId},{value},{tm})
</insert>
```

id：遵循 xml 唯一规范，为 putStorageMethod 的值，若为 update 方式则为 updateTestRealtimeData。

parameterType：需入库数据的类型，为 dataModelClassName 的值。

values ({regId},{value},{tm}) 中的 regId、value、tm 为 dataModelProperty 的属性值，该属性值意义与数据库字段一一对应。

6、数据召测

简要描述：

数据召测

请求 URL：

http://127.0.0.1:8081/topiot/service.api/sendcmd?serviceId=服务id®Id=设备id

请求方式：

post

参数：

Object

返回示例

```
{
    "reason" : "success",
    "result" : "数据已发送"
}
```

备注

该接口返回成功仅表示服务器端已将数据转发至设备，无法保证设备是否成功接收到指令。

7、实时数据获取

简要描述：

用户注册接口

请求 URL：

http://127.0.0.1:8081/topiot/data.api/getdata

请求方式：

post/get

参数：

参数名	必选	类型	说明
handler	是	string	应用层协议类名称
regId	是	string	设备 id
beginDate	否	date	数据起始日期
endDate	否	date	数据结束日期

备注

当起始日期大于结束日期时，将返回最新一条数据

返回示例

```
{
  "reason" : "success",
  "result" : {
    "0" : {
      "id" : "1",
      "regId" : "4056",
      "value" : "12.33",
      "time" : "1532702658119"
    },
    ...
    "12": {
      "id" : "11",
      "regId" : "4056",
```

```
        "value" : "19.33",  
        "time" : "1532702668521"  
    }  
}  
}
```